

Amendments to the Claims

1.-16. (Canceled)

17. (New) A method of translating data from a format of a data model of a first software component to a format of a data model of a second software component, the method comprising the steps of:

- creating a first schema comprising the data model of the first software component;
- integrating the first schema into a data wedge;
- creating a second schema comprising the data model of the second software component;
- integrating the second schema into the data wedge;
- populating the data model of the first software component; and
- translating a data element from the format of the data model of the first software component to the format of the data model of the second software component by the data wedge.

18. (New) The method of claim 17, further comprising the step of:  
triggering an event to notify the second software component of translated data element availability.

19. (New) The method of claim 17, further comprising the step of:  
reading the translated data element by the second software component.

20. (New) The method of claim 17, further comprising the step of:  
removing an obsolete data element from the data model of the first software component.

21. (New) The method of claim 17, further comprising the step of:  
creating an instance of the data wedge.

22. (New) The method of claim 17, wherein the first and second schemas further comprise a name of the data wedge.

23. (New) The method of claim 17, wherein integrating the first schema into the data wedge includes setting default data elements and data values for the data model of the first software component.

24. (New) The method of claim 17, further comprising the step of:  
modifying a data element in the data model of the first software component.

25. (New) A computer system for translating data from a format of a data model of a first software component to a format of a data model of a second software component, the system comprising:

a processor; and

a memory coupled to said processor, the memory having stored therein data and sequences of instructions which, when executed by said processor, cause said processor to:

create a first schema comprising the data model of the first software component;

integrate the first schema into a data wedge;

create a second schema comprising the data model of the second software component;

integrate the second schema into the data wedge;

populate the data model of the first software component; and

translate a data element from the format of the data model of the first software component to the format of the data model of the second software component by the data wedge.

26. (New) The system of claim 25, further comprising instructions which, when executed by said processor, cause said processor to:

trigger an event to notify the second software component of translated data element availability.

27. (New) The system of claim 25, further comprising instructions which, when executed by said processor, cause said processor to:

remove an obsolete data element from the data model of the first software component.

28. (New) The system of claim 25, further comprising instructions which, when executed by said processor, cause said processor to:

create an instance of the data wedge.

29. (New) The system of claim 25, further comprising instructions which, when executed by said processor, cause said processor to:

modify a data element in the data model of the first software component.

30. (New) The system of claim 25, wherein the first and second schemas further comprise a name of the data wedge.

31. (New) The system of claim 25, wherein the instructions causing the processor to integrate the first schema into the data wedge include instructions causing the processor to set default data elements and data values for the data model of the first software component.

32. (New) A computer implemented system for translating data from a format of a data model of a first software component to a format of a data model of a second software component, the system comprising:

a data wedge including a first schema of the first software component and a second schema of the second software component, the data wedge configured to translate a data element from the format of the data model of the first software component in accordance with the first schema to the format of the data model of the second software component in accordance with the second schema.

33. (New) The system of claim 32, wherein said data wedge is further configured to trigger an event to notify the second software component of translated data element availability.